

## **REMARKS**

These remarks are in response to the non-final Office Action dated May 6, 2005. Claims 1-111 are pending in the application.

In the non-final Office Action, the Examiner rejected claims 1-111 under 35 U.S.C. § 102(e) as being anticipated by or, in the alternative, under 35 U.S.C. § 103(a) as obvious over, U.S. Patent No. 6,226,642 to Beranek et al. ("Beranek"). Claims 1-111 were also rejected under 35 U.S.C. § 103(a) as being unpatentable over Beranek in view of U.S. Patent. No. 6,408,336 to Schneider et al. ("Schneider").

Each of the rejections from the non-final Office Action of May 6, 2005 is discussed below in connection with the various claims. Further, with this response, paragraph number 88 of the Specification has been amended in order to correct a typographical error; support for this amendment may be found in the Specification. No new matter has been added. Applicants respectfully request reconsideration of the application in light of the following remarks.

### **I. INDEPENDENT CLAIMS 1, 41, 50, 72, 90, AND 108**

#### **A. Rejections Under 35 U.S.C. § 102(e)**

Claims 1, 41, 50, 72, 90, and 108 were rejected under 35 U.S.C. § 102(e) as being anticipated by Beranek. Applicants submit that Beranek fails to anticipate claims 1, 41, 50, 72, 90, and 108 because Beranek fails to disclose all the limitations of these claims.

Beranek discloses a "... method [which] uses a client side HTTP caching proxy to intercept the Web document and then dynamically rewrite the document before it is displayed on the browser of the Web appliance." Beranek at Abstract.

Beranek fails to disclose claims 1, 41, 50, 72, 90, and 108 in their entirety. Particularly, Beranek at least fails to disclose "capturing said first data packet from said network prior to its reception by said first recipient," as claimed in claim 1; "intercepting said first data packet prior to receipt by said first recipient," as claimed in claim 41; and "a network interface operative to receive said first packet from said

source; a routing processor coupled with said network interface and operative to receive said first packet from said network interface and convey said first packet to said first destination,” as claimed in claim 50. Furthermore, Beranek also fails to at least disclose “a packet processor coupled with said router interface and operative to intercept a first packet prior to receipt by said router,” as claimed in claim 72; “a router coupled with said network and operative to route a first packet from a first source to a first destination; and a packet processor coupled with said router and operative to receive said first packet from said first source and process said first packet prior to routing by said router,” as claimed in claim 90; and “a traffic interceptor operative to selectively intercept said network traffic stream between said POP and said network prior to said network traffic stream reaching its intended destination; and a traffic modifier operative to modify said selectively intercepted traffic and reinsert said modified selectively intercepted traffic into said network,” as claimed in claim 108.

Beranek discloses instead that, “[c]onnectivity between the proxy and the browser is achieved using the sockets mechanism by configuring the browser to pass the HTTP requests to the proxy. To send an HTTP GET request, the browser creates a packet (including the URL and other information) and then opens a socket using the sockets mechanism. The packet is then sent to the IP address/port number to service the HTTP request. Thus, when the browser issues an HTTP GET request, it binds to the socket and sends the request. The request is then intercepted and processed by the proxy instead of being sent directly over the network. . . .” Beranek at col. 12, lns. 40-50.

Beranek, therefore, fails to disclose capturing a data packet from the network prior to its reception by the first recipient. Under Beranek, the proxy, having sent the request, is the intended recipient and receives every packet, reformats the Web page, and then passes the re-formatted page to the browser. *See* Beranek at FIG. 7. As disclosed in Beranek and as known in the art, the browser is aware that the proxy exists because the browser is configured to pass the HTTP requests to the proxy, which then sends them to the intended destination on behalf of the browser, indicating

itself as the source of the transmission to which the responses should be directed. Therefore, it is the proxy to which the responses to the requests are directed, and not the browser. *See Beranek at col. 3, ln. 65 – col. 4, ln. 8; col. 12, lns. 40-50.*

Under Beranek, when the requested packets come back, they are addressed to the proxy, not the browser. After the proxy receives the packets, it re-formats the Web data based on browser-supplied information, and then the proxy passes the packets to the browser. *See Beranek at Abstract; col. 3, ln. 65 – col. 4, ln. 8.* Thus, under Beranek, it is the proxy that is the intended recipient of the packets rather than the browser. Because the browser is configured to interact with the proxy, it is the proxy that is the first recipient, and therefore, the proxy is not “intercepting” the packets.

Intercept means “to receive (a communication or signal directed elsewhere) usually secretly.” Merriam-Webster’s Collegiate Dictionary 651 (11th ed. 2003). Beranek incorrectly uses the word “intercept.” Under Beranek, the proxy is the intended recipient of the data sent by the server. Therefore, the proxy is not “intercepting” the data. While, to a user, the browser is the ultimate intended recipient of the data contained within the data packets sent by the server, the actual communications are from the server to the proxy and then from the proxy to the browser, wherein the proxy is the intended recipient of the server and the browser is the intended recipient of the proxy. *See Beranek at Abstract; FIG. 7; FIG. 12; col. 3, lns. 5-63.* Beranek, as noted by the Examiner, fails to disclose a data packet being intercepted prior to receipt by the intended recipient, i.e. the proxy. *See 05/06/2005 Office Action at 3.*

Beranek discloses a non-transparent proxy. A proxy may be referred to as “non-transparent” when the browser is configured to work with the proxy and thus, “knows” of the proxy’s existence. Under this protocol, the proxy listens for packets from the server, and the browser listens for packets directed to it from the proxy. In other words, the browser is not listening for packets from the server since it knows that the proxy is the packet-handling intermediary.

On the other hand, a proxy may be referred to as “transparent” when the browser is not configured to communicate with the proxy, and the browser does not “know” of the proxy’s existence. Because the browser is not configured to know about the proxy, both it and the proxy listen for packets sent from the server.

Applicants submit that the decision between implementing a transparent and non-transparent proxy is not a “design choice,” as suggested by the Examiner. 05/06/2005 Office Action at 2. Implementation of a non-transparent proxy involves configuration of the browser so that it knows of the proxy’s existence and so that it will work with the proxy, whereas transparent proxy implantation involves hiding the proxy away from the browser so that it has no, and need not have, knowledge of the proxy’s existence.

Claims 1, 41, 50, 72, 90, and 108 are directed to a transparent device because the data packet is captured from the network prior to its reception by said first recipient. The proxy disclosed in Beranek, however, is non-transparent because the browser is configured to work with the proxy, and it knows of the proxy’s existence: “one or more client machines connected to the proxy server ‘discover’ the characteristics of their respective display platforms and provide this information to the proxy server.” Beranek at col. 3, ln. 65 – col. 4, ln. 8. Thus, under Beranek, the first recipient is the proxy, which listens for packets, edits the web content, and then passes them along to the browser. *See* Beranek at Abstract, col. 3 lns. 5-63. Although the browser might not realize the a page is served to it from the proxy’s cache, rather than sent from the server to the proxy (*see* Beranek at col. 9, lns. 25-44; col. 11, ln. 62 – col. 12, ln. 21), the browser is aware that the proxy is the packet-handling intermediary. *See* col. 3, ln. 65 – col. 4, ln. 2.

Beranek also fails at least to disclose “examining, selectively, a dynamically specified portion of said application data layer of said first data packet according to a second rule,” as claimed in claim 1; “analyzing, selectively, a dynamically specified portion of said application data according to a second rule,” as claimed in claim 41; and “selectively analyze a dynamically specified portion of said application data layer according to a second rule,” as claimed in claim 50. Beranek also fails to disclose

“apply a second function to a dynamically specified portion of said application data layer of said first packet and produce a second result,” as claimed in claim 72; and “second logic operative to analyze a dynamically specified portion of said application data layer of said first packet according to said second rule,” as claimed in claim 90.

Beranek instead discloses “[a]ccording to the present invention, the caching proxy includes the filtering mechanism 229 for receiving a Web document formatted according to HTML, identifying the HTML tags . . . , re-formatting the Web document by modifying one or more characteristics of original HTML, and then passing the modified Web document to the browser for display.” Beranek at col. 10, lns. 21-27.

Thus, Beranek fails to disclose selective analysis because, under Beranek, the proxy receives all packets in order to determine if it should reformat them. *See* Beranek at col. 10, lns. 21-27. Furthermore, Beranek does not use the word “filter” to indicate selectivity. Instead, Beranek uses the word “filter” to define a “mechanism [that] is [ ] used to reformat the Web document according to some given protocol . . . .” Beranek at Abstract; *see also* Beranek at FIG. 6. Moreover, Beranek cannot disclose selective analysis of a dynamically specified portion of the application data layer because the entire portion of the packet containing Web formatting information is always reviewed in its entirety in order to accurately determine what needs to be re-formatted. *See* Beranek at FIG. 5; FIG. 6.

Accordingly, because Beranek fails to disclose all the limitations of claims 1, 41, 50, 72, 90, and 108, Applicants respectfully request that the Examiner withdraw the rejections of these claims.

## **B. Rejections Under 35 U.S.C. § 103**

### **i. Rejections Under 35 U.S.C. § 103(a) as Being Obvious over Beranek**

Claims 1, 41, 50, 72, 90, and 108 were rejected under 35 U.S.C. § 103(a) as being obvious over Beranek. Applicants submit that claims 1, 41, 50, 72, 90, and 108 are not obvious over Beranek.

As discussed earlier, Beranek does not disclose a transparent proxy because the browser is configured to interact with the proxy. *See* Beranek at col. 12, lns. 40-

50. Contrary to Beranek's teaching, claim 1 claims "capturing said first data packet from said network prior to its reception by said first recipient," claim 41 claims "intercepting said first data packet prior to receipt by said first recipient," claim 50 claims "a network interface operative to receive said first packet from said source; a routing processor coupled with said network interface and operative to receive said first packet from said network interface and convey said first packet to said first destination," claim 72 claims "a packet processor coupled with said router interface and operative to intercept a first packet prior to receipt by said router," claim 90 claims "a router coupled with said network and operative to route a first packet from a first source to a first destination; and a packet processor coupled with said router and operative to receive said first packet from said first source and process said first packet prior to routing by said router," and claim 108 claims "a traffic interceptor operative to selectively intercept said network traffic stream between said POP and said network prior to said network traffic stream reaching its intended destination; and a traffic modifier operative to modify said selectively intercepted traffic and reinsert said modified selectively intercepted traffic into said network."

Beranek enables "a Web author to generate a single version of a Web page that may be displayed consistent across many different types of display system platforms. This is primarily because the proxy functions to intercept and re-format the Web document (or components thereof) and/or injects new control information for modifying how the document is displayed on the browser." Beranek at col. 14, lns. 34-41. Beranek also discloses "one or more client machines connected to the proxy server 'discover' the characteristics of their respective display platforms and provide this information to the proxy server." Beranek at col. 3, ln. 65 – col. 4, ln. 2.

It would not be obvious to one skilled in the art to add the missing limitations of claims 1, 41, 50, 72, 90, and 108. Under Beranek, the browser is configured to work with the proxy. *See* Beranek at col. 3, ln. 65 – col. 4, ln. 8. Thus, if the browser was not configured to work with the proxy, the browser would send its own HTTP GET requests and the responses that it receives may not be compatible with the browser, thus negating the purpose and functionality of the Beranek system. *See*

Beranek at col. 12, lns. 40-50. Furthermore, if the proxy was transparent, the browser would not be aware of the proxy in order to be able to pass to the proxy the characteristics of its display platform as required under the Beranek system. *See* Beranek at col. 3, ln. 65 – col. 4, ln. 8. Still further, there is no motivation to combine the missing limitations with Beranek since to do so would make Beranek inoperable. Therefore, because the Beranek system would not work with a transparent proxy, such a change would not be obvious to one skilled in the art.

Furthermore, it would not be obvious to one skilled in the art to add “selecting” or “analyzing” a “dynamically specified portion of said application data layer” to Beranek. Beranek focuses on “identifying the HTML tags” and “re-formatting the Web document.” *See* Beranek at col. 10, lns. 21-26. Beranek would be inoperable if it selectively analyzed a “dynamically specified portion of said application data layer” since in order for Beranek’s to work, it must review the entire portion of the packet containing Web formatting information and make changes to it based on the client’s settings. *See* Beranek at FIG. 5; FIG. 6. As such, adding the missing limitations to Beranek is not obvious to one skilled in the art.

**ii. Rejections Under 35 U.S.C. § 103(a) as Being Obvious Over Beranek in View of Schneider**

**1. Independent Claims 1, 41, 50, 72, and 90**

Claims 1, 41, 50, 72, and 90 were rejected under 35 U.S.C. § 103(a) as being obvious over Beranek in view of Schneider. Applicants submit that claims 1, 41, 50, 72, and 90 are not obvious over Beranek in view of Schneider because these references, alone or in combination, fail to disclose all of the elements of the claims.

Schneider discloses “[a] scalable access filter that is used together with others like it in a virtual private network to control access by users at clients in the network [sic] to information resources provided by servers in the network.” Schneider at Abstract. Schneider also discloses, “[a]ccess is permitted or denied according to of [sic] access policies which define access in terms of the user groups and information sets.” Schneider at Abstract.

As discussed above, Beranek fails to disclose “examining, selectively, a dynamically specified portion of said application data layer of said first data packet according to a second rule,” as claimed in claim 1; “analyzing, selectively, a dynamically specified portion of said application data according to a second rule,” as claimed in claim 41; and “selectively analyze a dynamically specified portion of said application data layer according to a second rule,” as claimed in claim 50. Beranek also fails to disclose “apply a second function to a dynamically specified portion of said application data layer of said first packet and produce a second result,” as claimed in claim 72; and “second logic operative to analyze a dynamically specified portion of said application data layer of said first packet according to said second rule,” as claimed in claim 90.

As understood in the art, and disclosed by Schneider, for the firewall to operate, the entire application data must be reviewed in order to accurately determine if access to the network should be granted. *See* Schneider at col. 3, ln. 60 – col. 4, ln. 6. Under Schneider, in order to perform an access check, the filter must go through a predetermined protocol: “all access filters 203 have a layered architecture. The bottommost layer is an Internet packet filter 2419 that deals only with Internet packet headers. Packet filter 219 reads the source and destination addresses in the Internet packet headers and applies a set of rules to them. As determined by the rules, it either accepts them, discards them, or routes them further in VPN 201. The rules also determine how the accepted packets are to be routed within access filter 203. The next layer of the architecture is service proxies 2427. The service proxies intercept traffic for services such as the World Wide Web and do access checking on the traffic. If access filter 203 provides the service itself or does access checking for a server that provides the service, IP filter 2419 sends packets intended for the service to a service proxy 2427 for the service. The service proxy uses access control database 301 to do protocol-level access checking for the service.” Schneider col. 27, lns. 20-36; *see also* Schneider at col. 29, ln. 63 – col. 30, ln. 34. Based on the protocol, the access filters must reference the tables that contain information regarding to what types of information one may have access: “[a]ccess filter 203 can



thus determine for a given user identification whether it identifies a user that has access, what kind of user identification it is, and therefore what trust level it has, and which user groups the user belongs to. User group tables 1301 thus contain all of the information needed for the user portion of an access policy 1108." Schneider at col. 30, Ins. 29-34. Because the entire packet must be reviewed, it does not selectively analyze a dynamically specified portion of the application data.

Furthermore, something is "dynamic" when it is "marked by usually continuous and productive activity or change." Merriam-Webster's Collegiate Dictionary 389 (11th ed. 2003). Under Schneider, the access tables are static and thus, the rules applied for packet-analysis are also static. Although the rules can be changed, such changes are not "dynamic." *See* Schneider at col. 7, Ins. 5-15. Instead, under Schneider, all changes must be carried out in the master copy of the policy and then propagated to all the other access filters: "[a]n administrative user may edit the local copy and changes are propagated to the other access filters in the virtual private network. One of the access filters has a master copy, and changes are first propagated to the master copy and the changed master copy is then propagated to all of the other access filters." Schneider at col. 7, Ins. 5-15; *see also* Schneider at col. 8, Ins. 63-67; col. 35, ln. 59 – col. 36, ln. 60. Thus, unlike in Applicants' claims, Schneider fails to selectively analyze a dynamically specified portion of the application data.

Accordingly, because Beranek in view of Schneider fails to disclose all limitations of claims 1, 41, 50, 72, and 90, Applicants respectfully request that the rejections be withdrawn.

## **2. Independent Claim 108**

Claim 108 was also rejected under 35 U.S.C. § 103(a) as being obvious over Beranek in view of Schneider. Applicants submit that claim 108 is not obvious over Beranek in view of Schneider because Schneider fails to fill Beranek's gaps.

As discussed above, Beranek fails to disclose "a traffic interceptor operative to selectively intercept said network traffic stream between said POP and said network prior to said network traffic stream reaching its intended destination; and a traffic

modifier operative to modify said selectively intercepted traffic and reinsert said modified selectively intercepted traffic into said network," as claimed in claim 108.

First, Schneider fails to disclose selective interception. Under Schneider, all packets much be checked for access at least once before the packet is permitted to be passed along to another access filter. *See* Schneider col. 48, lns. 16-28. Second, under Schneider, the IP filter of the access filter acts as a packet-traffic officer—it, *inter alia*, passes along permitted traffic, blocks non-permitted traffic, and drops non-permitted traffic. *See* Schneider at col. 39, lns. 2-20. The IP filter operates based on the access policies. *See* Schneider at col. 38, ln. 66 – col. 39, ln. 1. Under Schneider, a dropped packet is neither modified nor reinserted back into the network as claimed in claim 108. *See* Schneider at col. 33, lns. 54-57.

Accordingly, because Beranek in view of Schneider fails to disclose all limitations of claim 108, Applicants respectfully request that the rejections be withdrawn.

## **II. Dependent Claims 2-40, 42-49, 51-71, 73-89, 91-107, and 109-111**

Dependent claims 2-40, 42-49, 51-71, 73-89, 91-107, and 109-111 were rejected under 35 U.S.C. § 102(e) as being anticipated by or, in the alternative, under 35 U.S.C. § 103(a) as being obvious over Beranek. These claims were also rejected under 35 U.S.C. § 103 as being unpatentable over Beranek in view of Schneider.

Claims 2-40 are dependent on independent claim 1, claims 42-49 are dependent on independent claim 41, claims 51-71 are dependent on independent claim 50, claims 73-89 are dependent on independent claim 72, claims 91-107 are dependent on independent claim 90, and claims 109-111 are dependent on independent claim 108. As explained above, Beranek does not disclose all of the elements of independent claims 1, 41, 50, 72, 90, or 108. Also as explained above, the limitations not disclosed in Beranek are not obvious. Furthermore, these limitations are not obvious over Beranek in view of Schneider. Accordingly, all of the elements of claims 2-40, 42-49, 51-71, 73-89, 91-107, and 109-111, which depend from the independent claims, are not disclosed. Therefore, Applicants respectfully

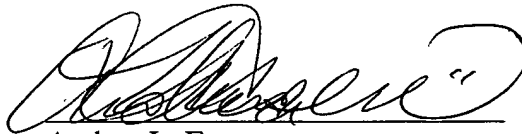
request the withdrawal of the rejections of claims 2-40, 42-49, 51-71, 73-89, 91-107, and 109-111 under 35 U.S.C. § 102(e) and § 103.

## CONCLUSION

In view of the foregoing remarks, Applicants submit that the pending claims are in condition for allowance. Applicants respectfully request reconsideration of the application. If there are any questions concerning this response, the Examiner is invited to phone the undersigned attorney at (312) 321-4200.

Respectfully submitted,

Dated: 7-13-05



Andrea L. Evensen  
Registration No. 56,531  
Attorney for Applicants

BRINKS HOFER GILSON & LIONE  
P.O. BOX 10395  
CHICAGO, ILLINOIS 60610  
(312) 321-4200